



# Open2Test Test Automation Framework for Selenium Web Driver - Quick Start Guide

Version 1.0

Apr 2013

## DISCLAIMER

*Verbatim copying and distribution of this entire article are permitted worldwide, without royalty, in any medium, provided this notice is preserved.*

**TABLE OF CONTENTS**

1	PURPOSE.....	3
2	ENVIRONMENT SETUP.....	4
3	FRAMEWORK AT GLANCE .....	5
3.	PROJECT SETUP IN ECLIPSE .....	7
4	USAGE OF KEYWORDS.....	14
5	TEST RESULTS FOR A KEYWORD-DRIVEN SCRIPT.....	15

## **1 PURPOSE**

This document explains the support settings and how to get started with keyword-driven scripting using Selenium WebDriver.

## 2 Environment Setup

The following jar files are required to run the Open2Test Selenium Webdriver Framework.

- jexcelapi\_2\_6\_12 or higher
- junit-4.9 or higher
- selenium-server-standalone-2.7.0 or higher

### 3 Framework at Glance

#### Input Files:

##### a. Selenium Utility Excel:

The TestSuite, TestScript, Object Repository and Report Folder location should be mentioned in Selenium\_Utility excel.

File\Folder Name	Location
Test Suite	D:/path/Test_Suite1.xls
Test Script	D:/path/
Object Repository	D:/path/TestData/ObjectRepository.xls
Summary Report	D:/path/Test Reports/
Screen Shot Report	D:/path/Test Reports/ScreenShot_Report/
Detailed Report	D:/path/Test Reports/Detailed_Report/

Note: The selenium utility excel file should be placed in

D:\Selenium\_Utility.xls. If you wish to place the utility file in some other location, you must update the path in the ReadUtilFile() function in the framework.

##### b. Object Repository Excel:

Collect the properties of objects and define it in Object\_Repository.xls like below

Object Name	Object Type	Parent	ObjectPath
Welcome to Open2Test.org	Page	0	name=
logo.jpg	Image	1	xpath=/html[1]/body[1]/div[1]/div[1]/img[1]
Home	Link	1	link=Home
About Us	Link	1	link=About Us
Tech Docs	Link	1	link=Tech Docs
News	Link	1	link=News
Roadmap	Link	1	link=Roadmap
Comment	Link	1	link=Comment
Support	Link	1	link=Support
Contact Us	Link	1	link=Contact Us
banner.jpg	Image	1	xpath=/html[1]/body[1]/div[1]/div[2]/div[1]/img[1]
Learn More	Link	1	link=Learn More
Learn More_1	Link	1	link=Learn More
Name	Textbox	1	id=name
emailID	Textbox	1	id=emailID
selectframework	ComboBox	1	xpath=//select
Agreement	Textbox	1	id=Agreement

Terms & Conditions	Link	1	link=Terms & Conditions
Input	Textbox	1	id=submit
new.gif	Image	1	xpath=/html[1]/body[1]/div[1]/div[2]/div[2]/div[2]/p[1]/img[1]
More news	Link	1	link=More news
Sitemap	Link	1	link=Sitemap
Privacy Policy	Link	1	link=Privacy Policy

#### c. Test suite Excel:

Mention the test scripts in the 'Test suite' excel in the order you wish to execute them.

Run	Test Scripts
r	Mantis_Create Project
r	Mantis_Report Issue
r	Mantis_View Issue and Delete Project
r	Mantis_Fail
r	Call Tariff Addition and Amendment

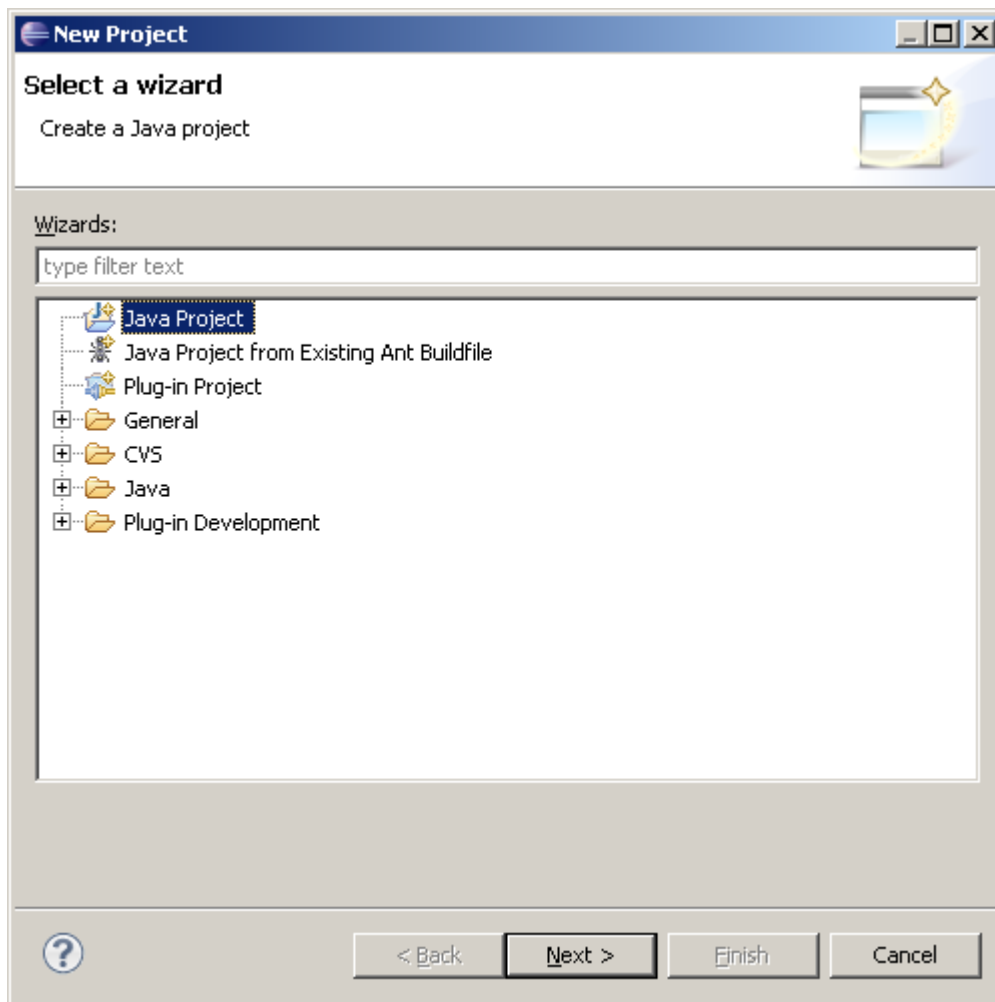
#### d. Test Script Excel

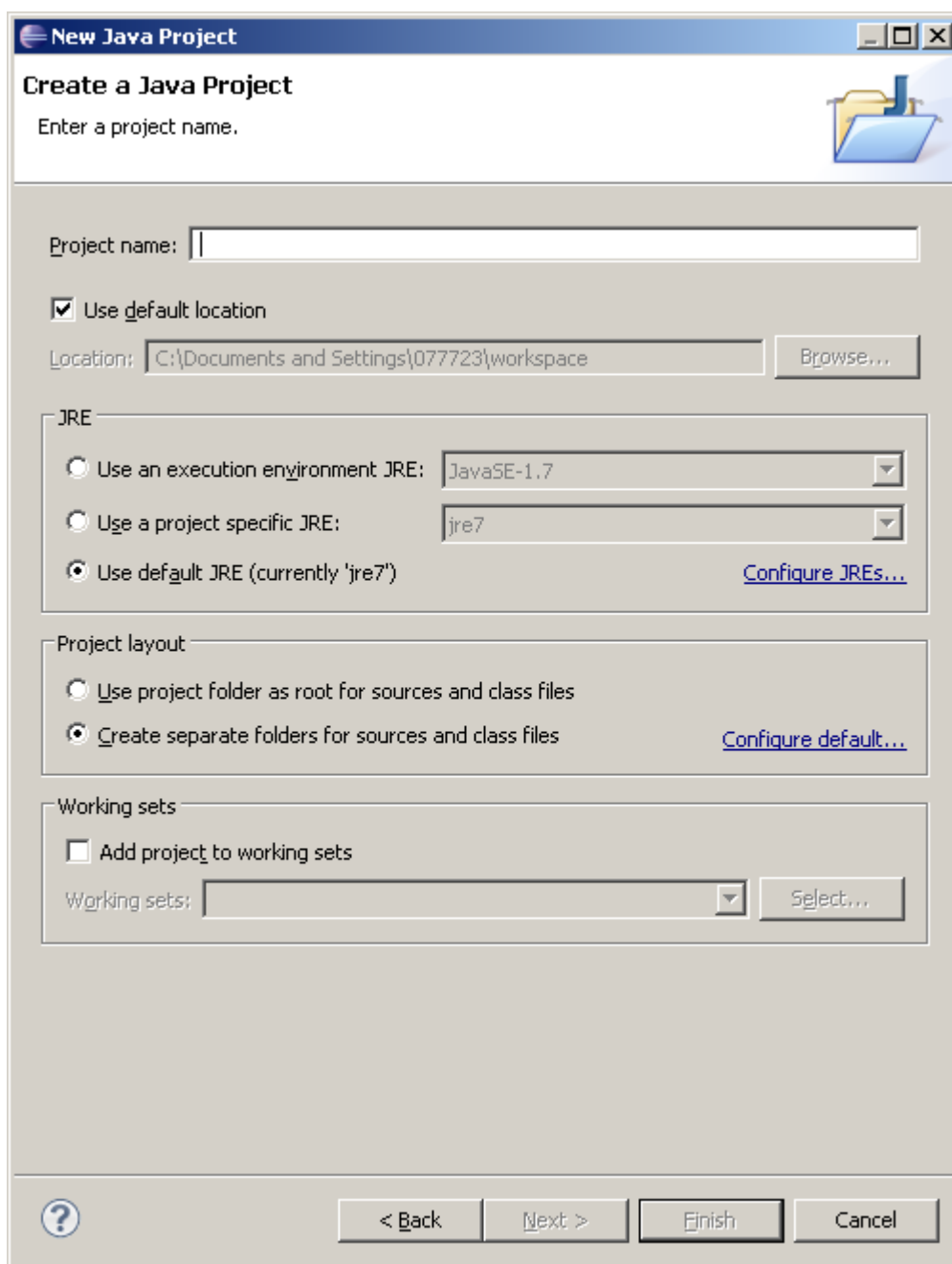
Write the keywords for the test script like below

Run	Keyword	ObjectDetails	Action	Action2	Comments
r	LaunchApp	<a href="http://www.open2test.org">www.open2test.org</a>			
r	importdata	<a href="D:\001_D_Files\path\TestData\TestData.xls">D:\001_D_Files\path\TestData\TestData.xls</a>	Sheet1		
r	loop	3			
r	perform	Textbox;name	set:dt_username		
r	perform	Textbox;emailID	set:dt_email		
r	check	Textbox;emailID	enabled:true		
	call				
r	perform	ComboBox;selectframework	select:Web Framework (V2) for Selenium		
r	perform	Textbox;Agreement	click		
	perform	Link;Terms & Conditions	click		
r	perform	Button;input	click		
r	screenshot				
r	Endloop				

### 3. Project Setup in Eclipse

Create a Java project in Eclipse





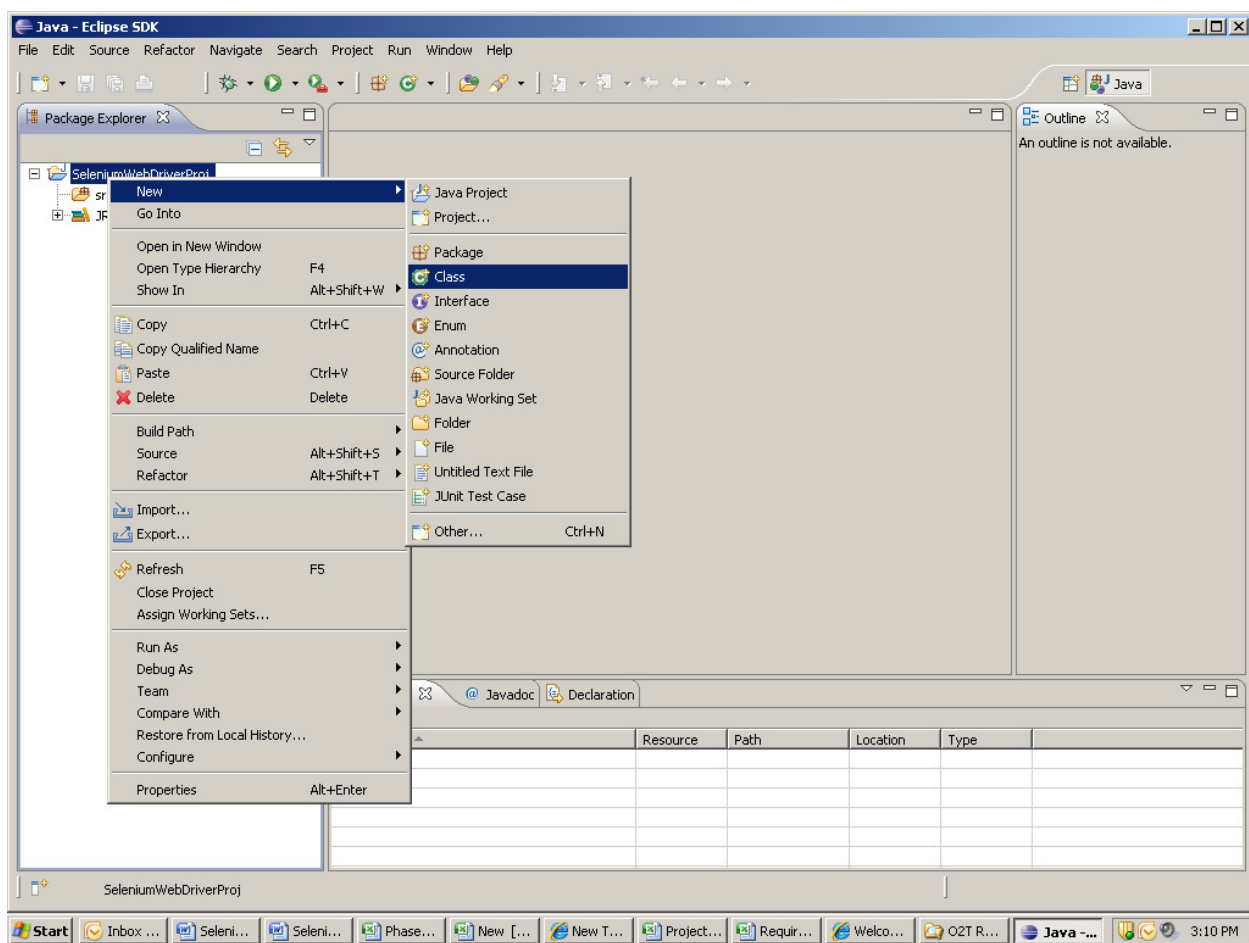
The image shows the 'New Java Project' dialog box in an IDE. It has a title bar with the text 'New Java Project' and standard window controls. The main area is titled 'Create a Java Project' and contains the following sections:

- Project name:** A text input field with a cursor.
- Location:** A text input field containing 'C:\Documents and Settings\077723\workspace' and a 'Browse...' button.
- JRE:** A section with three radio buttons:
  - ☐ Use an execution environment JRE: JavaSE-1.7
  - ☐ Use a project specific JRE: jre7
  - ☒ Use default JRE (currently 'jre7')A 'Configure JREs...' link is to the right.
- Project layout:** A section with two radio buttons:
  - ☐ Use project folder as root for sources and class files
  - ☒ Create separate folders for sources and class filesA 'Configure default...' link is to the right.
- Working sets:** A section with a checkbox 'Add project to working sets' and a 'Working sets:' dropdown menu with a 'Select...' button.

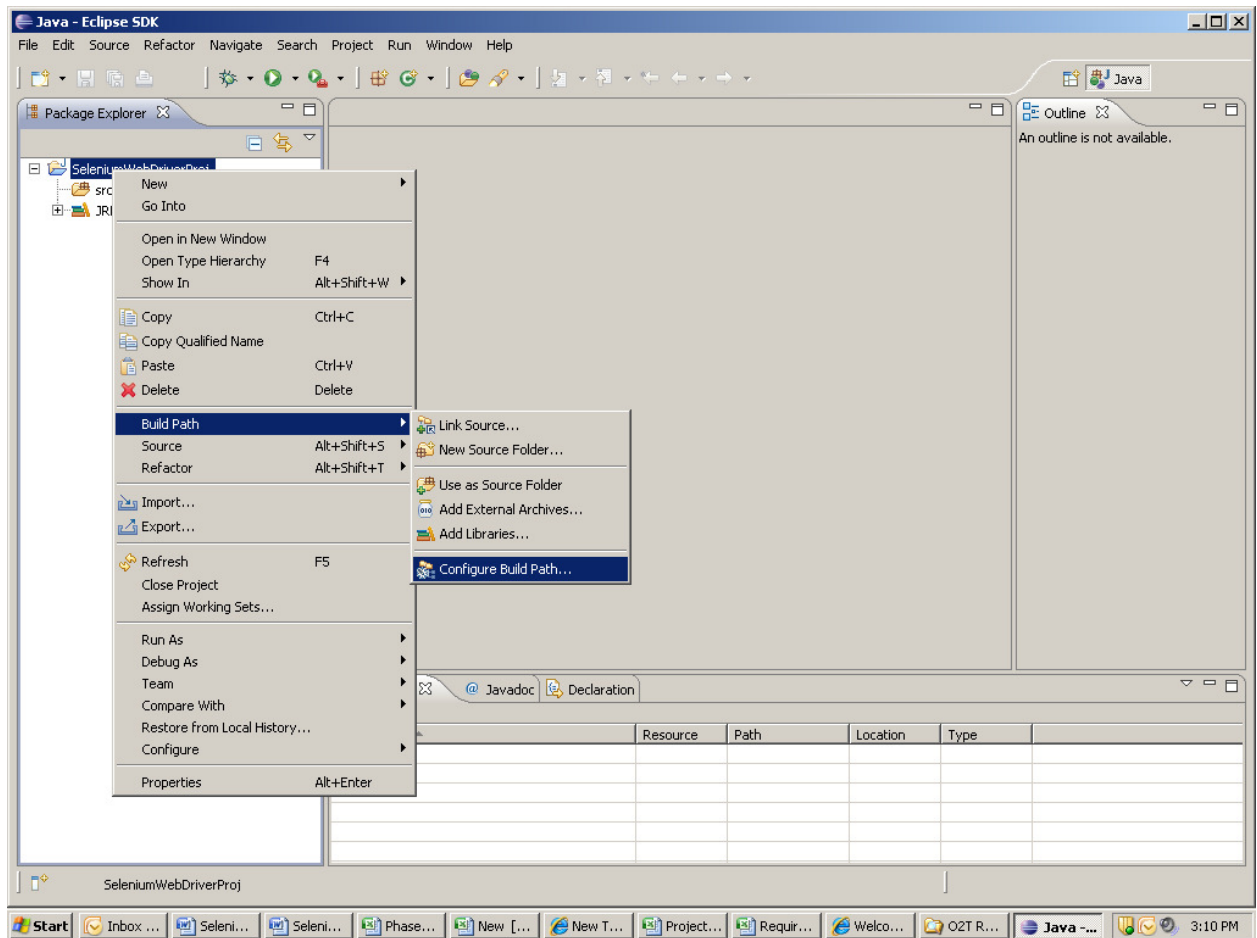
At the bottom, there is a help icon (question mark) and four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

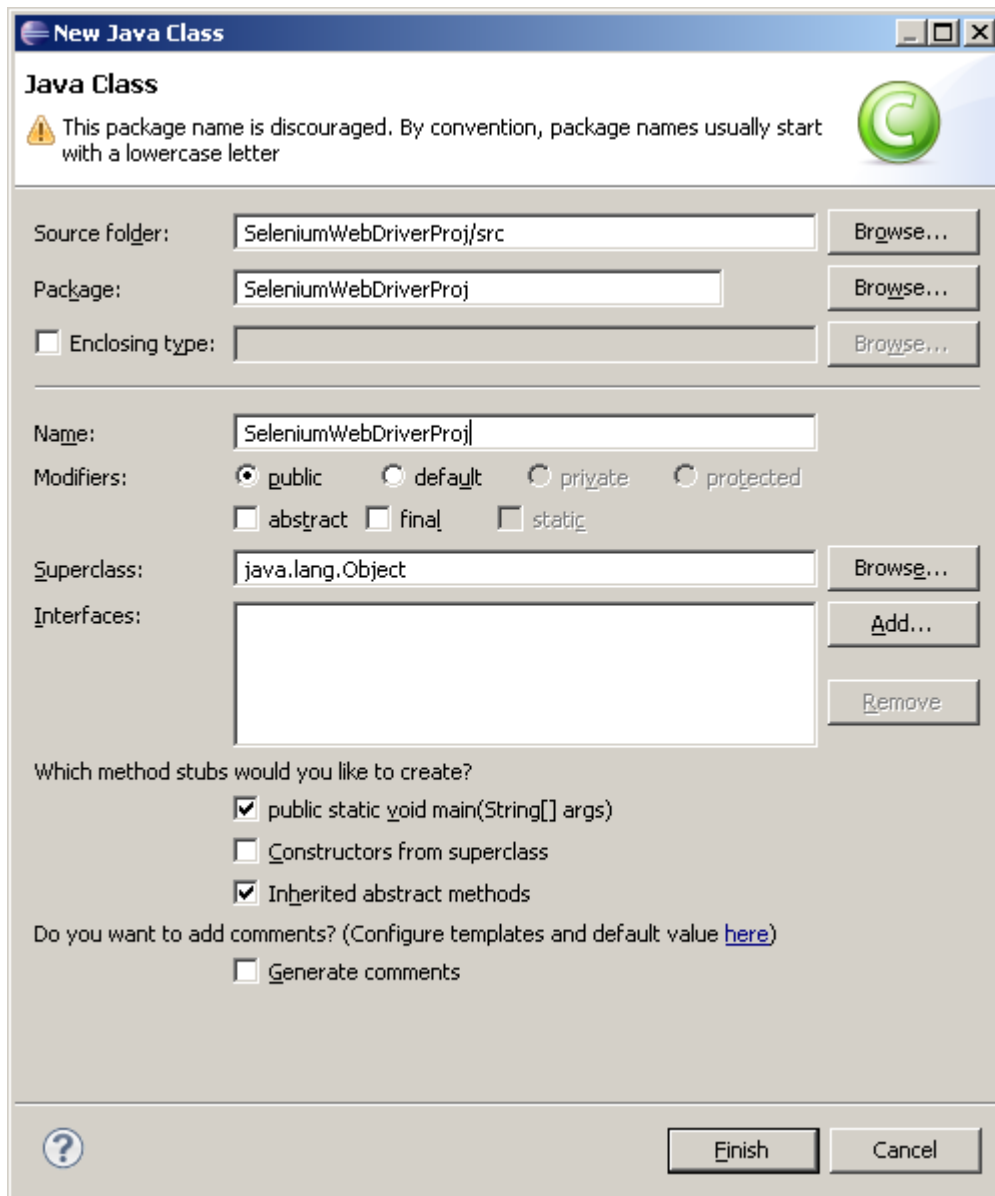
**Note:** Please note that JavaSE-1.7 has to be used





Add the required jar files to the library:





The image shows a 'New Java Class' dialog box from an IDE. It has a title bar with standard window controls. The main area is titled 'Java Class' and contains a warning icon and text: 'This package name is discouraged. By convention, package names usually start with a lowercase letter'. Below this, there are three input fields with 'Browse...' buttons: 'Source folder:' (SeleniumWebDriverProj/src), 'Package:' (SeleniumWebDriverProj), and 'Enclosing type:' (empty). The 'Name:' field contains 'SeleniumWebDriverProj'. Under 'Modifiers:', there are radio buttons for 'public' (selected), 'default', 'private', and 'protected', and checkboxes for 'abstract', 'final', and 'static'. The 'Superclass:' field contains 'java.lang.Object' with a 'Browse...' button. The 'Interfaces:' section has an empty list box with 'Add...' and 'Remove' buttons. A section titled 'Which method stubs would you like to create?' has three checked options: 'public static void main(String[] args)', 'Constructors from superclass', and 'Inherited abstract methods'. Below this, a question 'Do you want to add comments? (Configure templates and default value [here](#))' has an unchecked 'Generate comments' checkbox. At the bottom are 'Finish' and 'Cancel' buttons, and a help icon on the left.

**New Java Class**

**Java Class**

⚠ This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder: SeleniumWebDriverProj/src Browse...

Package: SeleniumWebDriverProj Browse...

☐ Enclosing type: Browse...

Name: SeleniumWebDriverProj

Modifiers: ☒ public ☐ default ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ public static void main(String[] args)

☐ Constructors from superclass

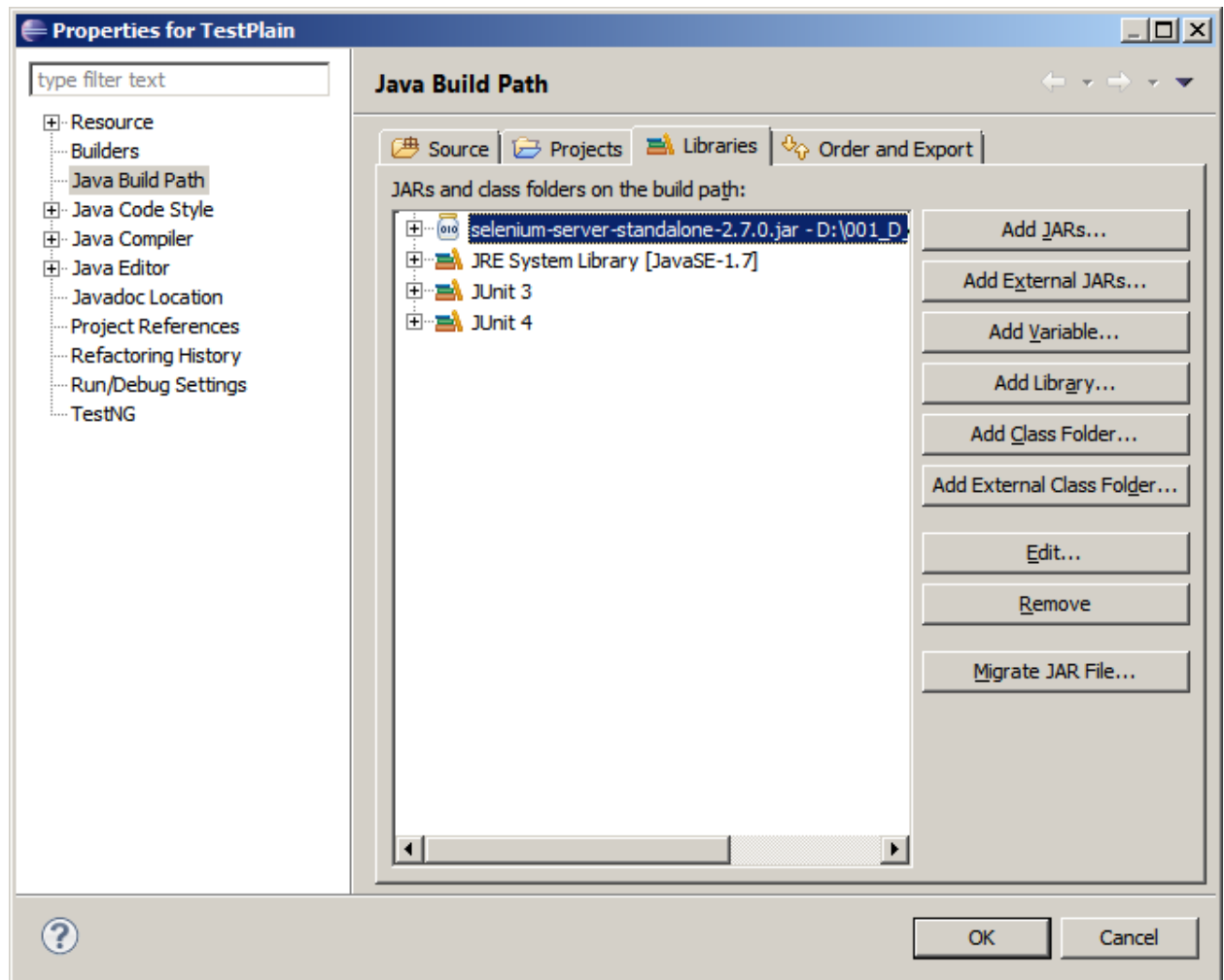
☒ Inherited abstract methods

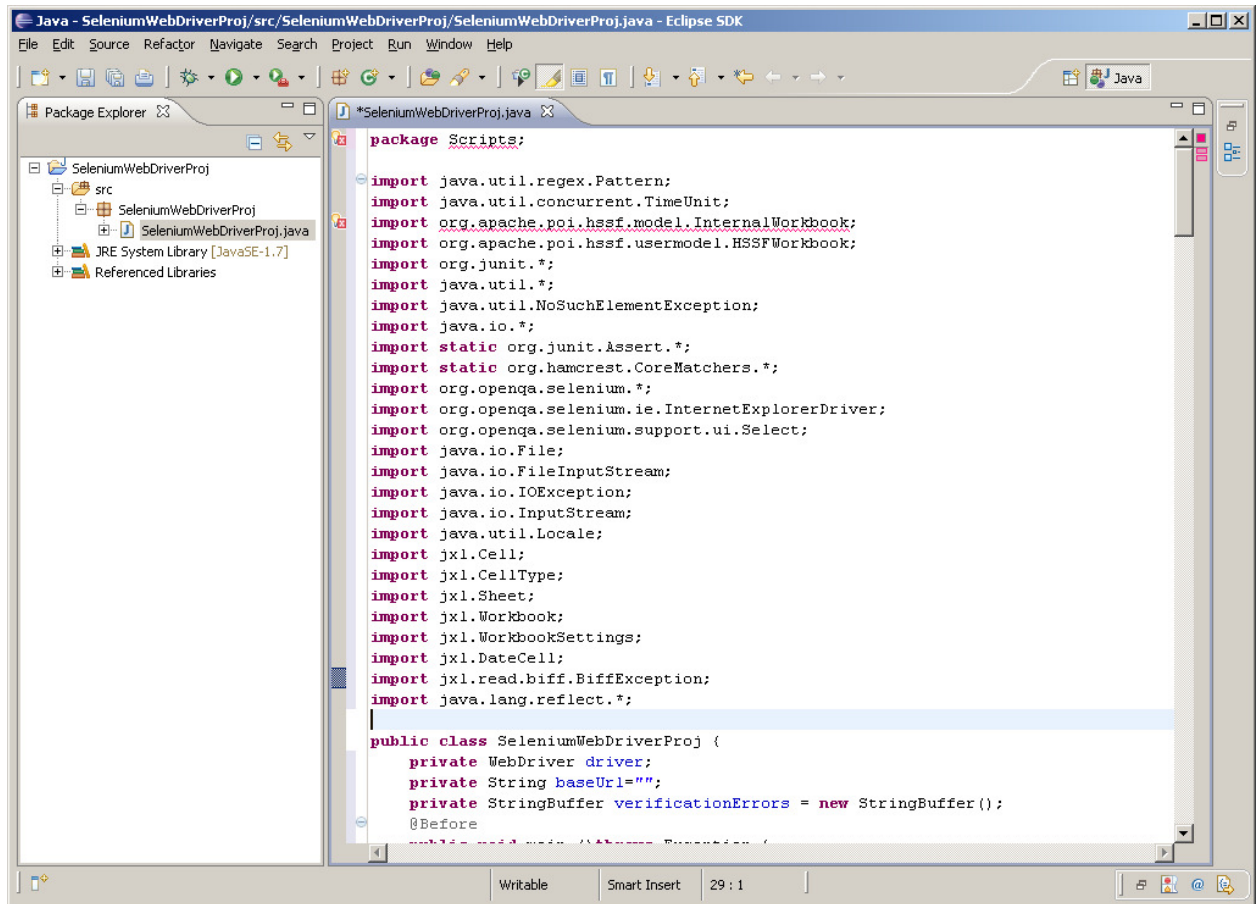
Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

Finish Cancel

Remove the contents of the new class and Copy the framework into the new class file





The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: SeleniumWebDriverProj, src, SeleniumWebDriverProj, SeleniumWebDriverProj.java, JRE System Library [JavaSE-1.7], and Referenced Libraries. The main editor window shows the code for SeleniumWebDriverProj.java. The code includes a package declaration, a list of imports, and the start of a public class SeleniumWebDriverProj with private fields for WebDriver, baseUrl, and verificationErrors.

```
package Scripts;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.apache.poi.hssf.model.InternalWorkbook;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.junit.*;
import java.util.*;
import java.util.NoSuchElementException;
import java.io.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.support.ui.Select;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Locale;
import jxl.Cell;
import jxl.CellType;
import jxl.Sheet;
import jxl.Workbook;
import jxl.WorkbookSettings;
import jxl.DateCell;
import jxl.read.biff.BiffException;
import java.lang.reflect.*;

public class SeleniumWebDriverProj {
    private WebDriver driver;
    private String baseUrl="";
    private StringBuffer verificationErrors = new StringBuffer();
    @Before
    public void setUp() throws Exception {
```

## **4 Usage of Keywords**

The keywords for the test scenario should be entered in the Sheet1 of the Microsoft Excel placed in the Test Scripts folder. The syntax for the keywords can be referred from Selenium Keywords Reference Dictionary Document.

## 5 Test Results for a Keyword-Driven Script

Test execution results can be viewed and analyzed as soon as the run session ends. To access the test results, go to the Test\_Reports folder customized using the test automation framework. Two folders will be available: one showing the summary report for test suite execution, and another folder called Detailed\_Report that displays the detailed step-wise test results for each test script. A screenshot will be available for the failure scripts under ScreenShot\_Report.

---

**COPYRIGHT**

*This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.*

*This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.*

*You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.*